Classification of Urdu News Articles Using Machine Learning: A Comparative Analysis in Low-Resource Language Processing

M. Shafay Tanveer Hassan(26100057), Salaar Masood(26100149), M. Haseeb(26100253), Hamza Iqbal(26100130), M. Mustafa(26100038)

Abstract

This report presents a comprehensive study on classifying Urdu news articles from Pakistan into five distinct categories: Entertainment, World, Business, Sports, and Science & Technology. Using a dataset curated from prominent Urdu news channels, we explored and compared the performance of various machine learning models, including Neural Networks (NN), Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Support Vector Machines (SVM). Among these, Neural Networks, Logistic Regression, and Multinomial Naïve Bayes emerged as the top-performing models, offering the highest accuracy and consistency in classification. The results demonstrate the effectiveness of these models in handling multilingual text classification tasks and highlight their potential for further applications in natural language processing (NLP) for low-resource languages like Urdu.

1 Introduction

The rapid growth of digital news platforms in Pakistan has generated an immense volume of text data in Urdu, the country's national language. Categorizing this data is crucial for efficient information retrieval and content organization, especially given the linguistic complexities and lack of resources for Urdu text processing. This project addresses the challenge of classifying Urdu news articles into predefined categories—Entertainment, World, Business, Sports, and Science & Technology—using machine learning techniques.

Machine learning has revolutionized text classification tasks by automating the categorization process and reducing reliance on manual sorting. Despite extensive research in text classification for widely spoken languages like English, Urdu remains underrepresented in the domain. This study aims to bridge that gap by leveraging traditional and modern machine learning models.

Our initial experimentation included models such as Neural Networks, Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Support Vector Machines. These models were evaluated on a dataset comprising Urdu news articles from diverse sources. Through rigorous performance analysis, we identified Neural Networks, Logistic Regression, and Multinomial Naïve Bayes as the most effective classifiers for this task. This report provides insights into the preprocessing of Urdu text, model selection, training processes, and evaluation metrics used to achieve optimal results. The findings underscore the viability of applying machine learning techniques to address the challenges of multilingual text classification, with specific emphasis on Urdu. We first delve into our preprocessing and then to our testing, training, and fine-tuning of models.

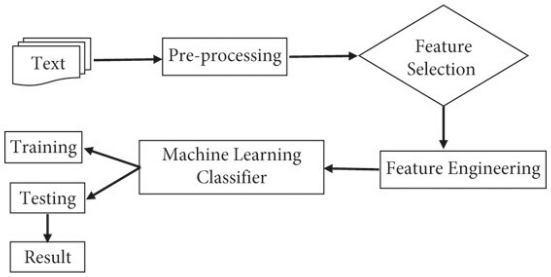

Figure 1: Project Workflow

2 Dataset Curation and Preprocessing

The dataset curation process was a critical component of our Urdu news classification project, aimed at ensuring high-quality data for machine learning models. Due to the lack of pre-existing datasets tailored to Urdu news, a comprehensive web-scraping and preprocessing pipeline was developed. Four prominent news websites were selected as reliable sources: *Express News*, *Geo News*, *Jang*, and *Samaa TV*. A prebuilt scraper was utilized for

Express News, while custom scripts were created for the other sources to standardize data extraction. Initially, 2,043 articles were collected and distributed as follows: 500 from *Express News*, 300 from *Geo News*, 750 from *Samaa TV*, and 493 from *Jang*. The dataset size was then increased to 2,535 articles with the following distributions: 749 from *Express News*, 300 from *Geo News*, 995 from *Samaa TV*, and 491 from *Jang*. This was done to improve model performance and data coverage by scraping additional data from the same sources.

The preprocessing and cleaning stage tailored the dataset to meet the specific requirements of various machine learning models. For Logistic Regression and Multinomial Naive Bayes, numerical data was excluded to focus on linguistic patterns. The preprocessing steps included the removal of punctuation, English words, extra spaces, and diacritics, ensuring standardization. Only Urdu script was retained, while stopwords—frequent yet insignificant Urdu words—were removed using a Kaggle dataset of 517 high-frequency words. Text normalization was applied using the UrduHack library, eliminating duplicate entries and empty rows. This clean dataset was saved as `cleaned_articles_without_numbers_v2.csv`. For Neural Networks, the preprocessing pipeline was adjusted to retain numerical data, as it could provide valuable context. The cleaning steps—such as punctuation, English words, diacritics, and stopword removal—were applied similarly while numbers were preserved. Duplicate entries and empty rows were also removed, and this dataset version was saved as `cleaned_articles_with_numbers_v2.csv`.

After completing the preprocessing and cleaning steps, the dataset was refined to a final size of 2,501 articles, ensuring comprehensive data coverage and high quality. This standardized dataset is well-suited for training and testing machine learning models, enabling robust and accurate performance.

3 Sklearn Implementations

To evaluate the performance of various machine learning algorithms for classifying Urdu news articles, we utilized the implementations provided by the scikit-learn (sklearn) library. Scikit-learn is a robust library for machine learning in Python that offers a wide range of tools for classification, regression, and clustering. We experimented with multiple models, including K-Nearest Neighbors

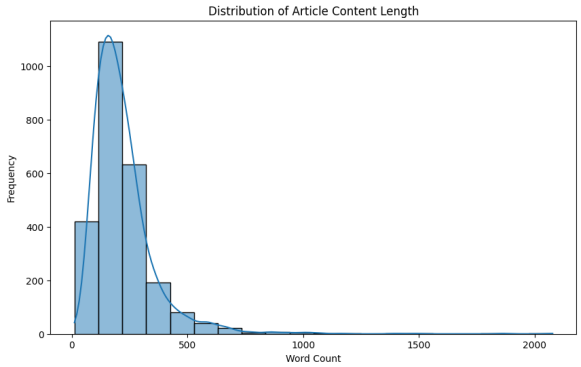

Figure 2: Articles Distribution

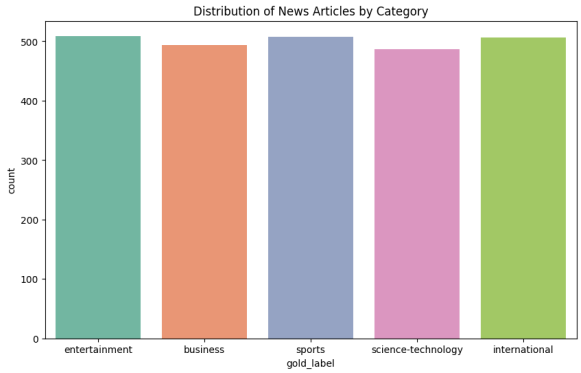

Figure 3: Article frequencies

(KNN), Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Support Vector Machines (SVM). The performance of these models was assessed based on their accuracy, as shown in Figure 4.

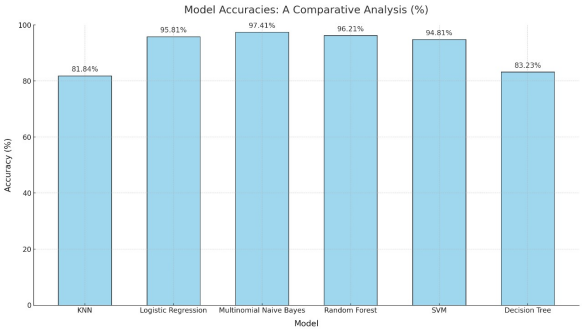

Figure 4: Performance Comparison of Classification Models on Sklearn

Logistic Regression and Multinomial Naïve Bayes stood out as the top-performing models, both achieving an impressive accuracy of 97%. Random Forest and SVM also delivered strong results, with accuracies of 94% and 93%, respectively, while KNN lagged behind at 79%. Given the superior performance of Logistic Regression and Multino-

mial Naïve Bayes, we chose to focus our efforts on these models for further analysis and optimization, as they demonstrated the most promise for accurately classifying Urdu text.

4 Manual Implementations

In this section, we delve into the manual implementations of the three selected models: Logistic Regression, Multinomial Naïve Bayes, and Neural Networks. Each model's methodology, findings, and limitations are discussed below.

4.1 Methodology

Logistic Regression was implemented using a Softmax Regression framework to handle the multiclass classification problem. The model employed L2 regularization to prevent overfitting and was trained on mini-batches of size 32 over 5000 iterations. The learning rate and regularization strength were fine-tuned to achieve optimal performance, with a learning rate of 0.001 and regularization strength of 0.03, yielding the best results. Dropout rates of 0.2 were tested and effectively prevented overfitting without losing essential features.

Multinomial Naïve Bayes utilized a bag-of-words representation to transform the text data into high-dimensional vectors. The training phase involved calculating log prior and conditional log probabilities with Laplace smoothing to handle unseen features in the test data. The model's probabilistic framework ensured computational efficiency and suitability for the discrete dataset. It predicted the class C_k for a given input $\mathbf{x}$ by maximizing the posterior probability, calculated as:

$$P(C_k \mid \mathbf{x}) = \frac{P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)}{\sum_{j=1}^{m} P(C_j) \prod_{i=1}^{n} P(x_i \mid C_j)}.$$

This approach allowed the model to focus on word frequency patterns critical for classification, leveraging the simplicity and efficiency of the bag-of-words method.

Neural Network was designed and implemented using PyTorch. The architecture consisted of an input layer, three hidden layers with 512, 256, and 128 units, and an output layer mapping to the five categories. Dropout layers with rates of 90% after the first hidden layer and 10% after the second were used to combat overfitting. The Adam optimizer with a learning rate of 0.001 dynamically adjusted training and CrossEntropyLoss was used for loss computation. Training data was processed in batches to maximize GPU utilization.

We also incorporated **UrduHack**, a specialized library for processing Urdu text, to evaluate our models further. We performed several preprocessing steps using its features, including stopword removal and handling join words. In Urdu text, "join words" refer to combinations of auxiliary verbs or particles with the main verb that are often written as separate words in text. For example, a phrase like "kar diya" (meaning "did") may be written with a space separating the components. UrduHack provides functionality to join such components into a single token, e.g., converting "kar diya" into "kardiya." This process ensures that these semantically cohesive units are treated as a single entity during text representation, potentially aligning better with the natural linguistic structure of Urdu.

Despite implementing these preprocessing steps, including the handling of join words, the modifications did not significantly improve the models' performance. The anticipated benefits of a more cohesive text representation were overshadowed by other limitations inherent in the dataset or the models' inability to effectively leverage these changes. Consequently, we decided to proceed without incorporating this preprocessing approach.

4.2 Findings

Logistic Regression achieved a strong accuracy of 96.41%, showcasing its robustness in extracting linear patterns from the data. The model achieved uniformly high evaluation metrics, with precision, recall, and F1-scores at 0.96, reflecting balanced performance across all categories. Its rapid convergence during training underscored the effectiveness of the chosen hyperparameters. The simplicity and reliability of Logistic Regression made it a solid choice for our dataset.

Multinomial Naïve Bayes outperformed the other models, achieving the highest accuracy of 98.00%. The model demonstrated uniformly high precision, recall, and F1-scores, all at 0.98, highlighting its ability to generalize effectively across the dataset. By leveraging the frequency distributions of words, it efficiently captured key features for classification. Furthermore, Laplace smoothing enabled it to gracefully handle sparse data and unseen features. This, combined with its lightweight computation, made it an ideal candidate for classi-

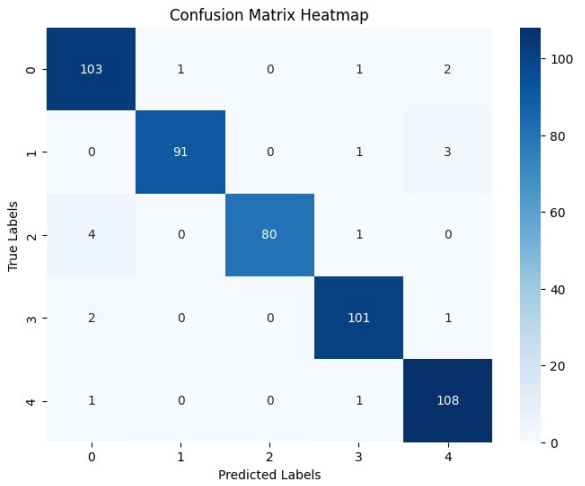

Figure 5: Heatmap for Logistic Regression

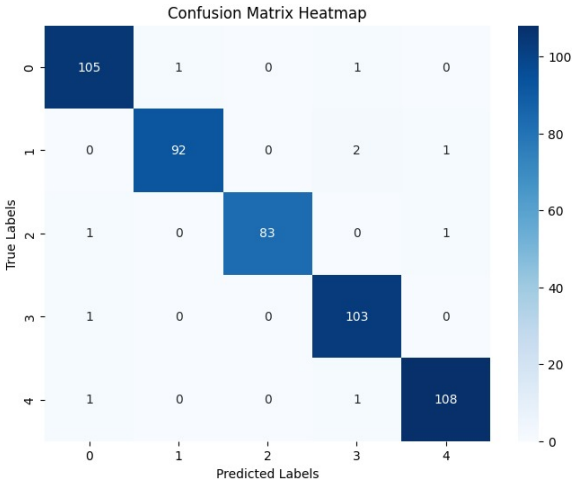

Figure 6: Heatmap Multinomial Naive Bayes

fication.

Neural Network achieved an accuracy of 97.21%, highlighting its ability to capture both linear and non-linear patterns in the data. The model demonstrated strong performance across all metrics, with precision, recall, and F1-scores slightly lower than Multinomial Naïve Bayes at 0.9728, 0.9721, and 0.9722, respectively. The use of progressively smaller hidden and dropout layers allowed the model to focus on essential features while mitigating overfitting. Despite requiring more computational resources, the Neural Network proved to be a versatile model, capable of tackling complex data relationships and delivering competitive results.

The superior performance of Multinomial Naïve Bayes can be attributed to its simplicity and ability to effectively utilize the probabilistic relationships in word frequencies, making it particularly well-suited for text classification tasks involving structured, categorical data. While efficient and balanced, Logistic Regression fell behind due to its reliance on linear patterns, which limited its flexibility for capturing more nuanced relationships in the data. On the other hand, the Neural Network demonstrated strong performance but slightly underperformed compared to Multinomial Naïve Bayes due to its sensitivity to training data quality and increased computational complexity. While its ability to model non-linear patterns was advantageous, it was not as crucial for this dataset, where simpler probabilistic techniques like those used by Naïve Bayes proved more effective.

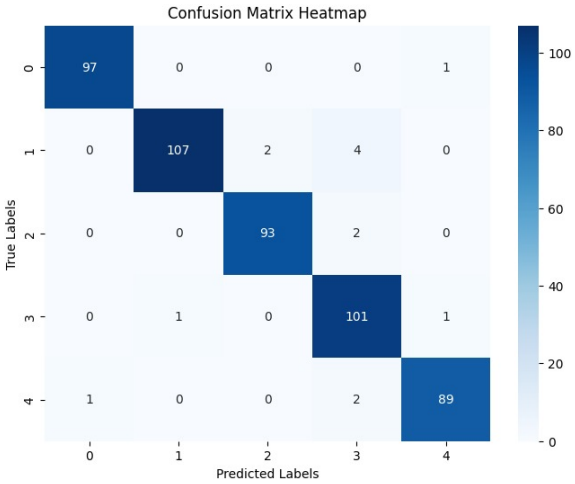

Figure 7: Heatmap for Neural Network

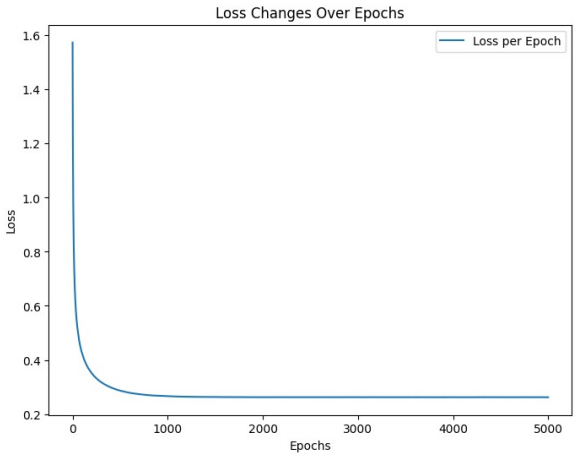

Figure 8: Logistic Loss

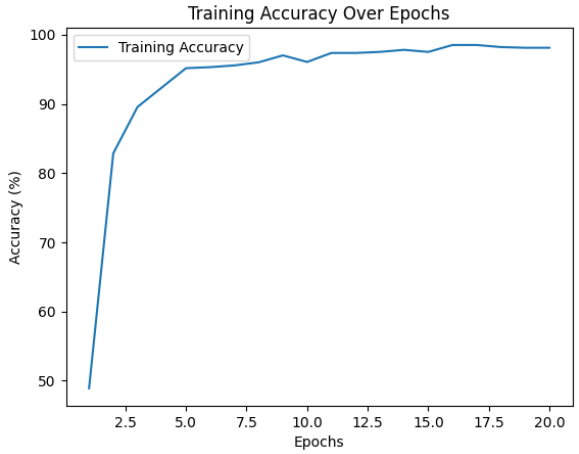

Figure 9: Neural Accuracy over time

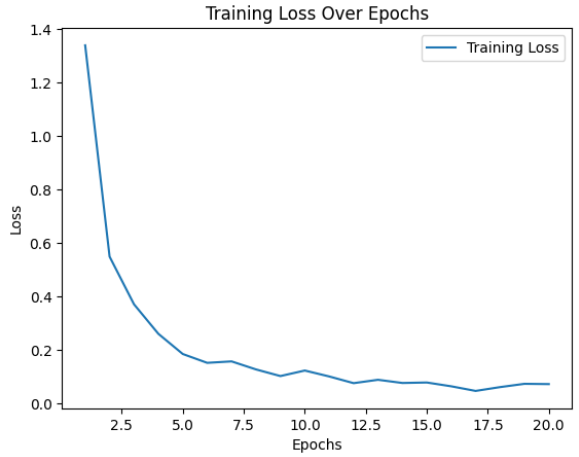

Figure 10: Neural Loss over time

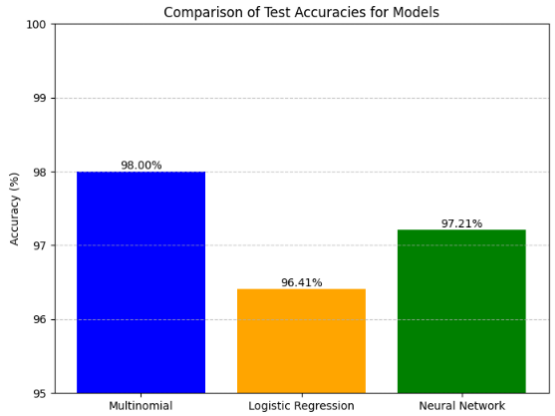

Figure 11: Performance Comparison of Classification Models

Model	Test Accuracy	Precision	Recall	F1-Score
Logistic Regression	96.41%	0.96	0.96	0.96
Multinomial Naïve Bayes	98.00%	0.98	0.98	0.98
Neural Network	97.21%	0.9728	0.9721	0.9722

Table 1: Performance Metrics of Selected Models

4.3 Limitations

The models showed varying levels of success, each with distinct limitations. The Neural Network, while effective in data representation, suffered from an excessively high dropout rate (0.9), limiting its ability to capture subtle patterns. Its performance was also dependent on high-quality preprocessing and struggled with inconsistencies in the test data, affecting its real-world applicability. Additionally, it required significant computational resources, raising concerns about scalability.

The Logistic Regression model was hindered by its assumption of linear patterns, making it less effective at capturing complex relationships. It also struggled with class imbalance, favoring the majority class, and required extensive training iterations to optimize parameters, highlighting the need for better feature selection and class-balancing techniques.

The Multinomial Naive Bayes model performed reasonably but was limited by its assumption of word independence, sacrificing important contextual information and reducing its ability to differentiate nuanced categories. This approach, while faster, restricted the model's generalization to complex patterns.

A key limitation of the study, time, led to a significantly smaller dataset than practical, which affected the models' performance and stability. Expanding the dataset further would have provided more robust learning patterns and addressed class imbalance. Additionally, the lack of test diversity hindered the models' ability to handle noisy or inconsistent data, a common challenge in real-world classification tasks. Expanding the dataset and optimizing hyperparameters for specific dataset characteristics could improve model robustness and performance.

5 Conclusion

This study highlights the potential of machine learning in tackling the unique challenges of classifying Urdu news articles, a task often overlooked due to the lack of resources for low-resource languages. By curating a robust dataset and testing multiple

machine learning models, we found that Multinomial Naïve Bayes stood out as the most accurate and efficient classifier, closely followed by Neural Networks and Logistic Regression. These results showcase the power of both simple probabilistic methods and more advanced neural models in understanding the nuances of Urdu text.

While the results are promising, there's room for improvement. A larger and more diverse dataset could help the models handle real-world complexities better, and exploring newer approaches like transformer-based models could unlock even greater potential. Expanding on this work could make organizing and accessing Urdu content easier, benefiting readers and researchers alike.

References

[1] Analytics Vidhya. (2021, December). Text classification of news articles. *Analytics Vidhya*. Retrieved from https://www.analyticsvidhya.com/blog/2021/12/text-classification-of-news-articles/

[2] GitHub. UrduHack Tokenization Module. *GitHub Repository*. Retrieved from https://github.com/urduhack/urduhack/blob/master/urduhack/tokenization/words.py

[3] Arshad, U., Qureshi, M. A., & Waqas, S. (2022). Urdu news content classification using machine learning algorithms. *Lahore Garrison University Journal of Computer Science & Information Technology*. Retrieved from https://lgurjcsit.lgu.edu.pk/index.php/lgurjcsit/article/view/274

[4] Farooq, A., Mumtaz, S., & Naz, F. (2022). Urdu news classification: An empirical study using machine learning techniques. *ResearchGate*. Retrieved from https://www.researchgate.net/publication/366703362_Urdu_News_Classification_An_Empirical_Study_Using_Machine_Learning_Techniques

[5] Rasheed, I., Zafar, M., & Shaikh, F. (2019). Urdu text classification: A comparative study using machine learning techniques. In *Proceedings of IEEE Xplore*. Retrieved from https://ieeexplore.ieee.org/document/8847044

[6] Abbas, S. Z., Mujtaba, M. N., & Rashid, T. (2022). Urdu news article recommendation model using natural language processing techniques. *arXiv preprint*. Retrieved from https://arxiv.org/abs/2206.11862

[7] Javed, T. A., Aslam, M., & Ali, A. (2021). Hierarchical text classification of Urdu news using deep neural network. *arXiv preprint*. Retrieved from https://arxiv.org/abs/2107.03141